Modelling Security Threats of Live Streaming Monetization Pipelines

Emil Eminov [1] and Stephen Flowerday [2,*]

[1] School of Cyber Studies, University of Tulsa, Tulsa, USA
[2] Department of Information Systems and Security, Augusta University, Augusta, USA
Email: eae5331@utulsa.edu (E.E.); sflowerday@augusta.edu (S.F.)
*Corresponding author

***Abstract*—Live-streaming platforms increasingly turn real-time payments, including cheers, Bits, gifts, and highlighted chat, into public signals of status and support. This article examines anomalous cheering as a security problem across the full monetization pipeline, asking how unusual spikes in volume, value, timing, or attribution can be related to system assets, trust boundaries, and controls. We develop a reference data-flow model of a monetized cheering pipeline and apply Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) to each system element across identity, payments, wallet and ledger services, event streaming, anomaly detection, and trust and safety operations. A simple ordinal Damage Potential, Reproducibility, Exploitability, Affected Users, and Discoverability (DREAD) rubric is used for prioritization, with a worked example showing how one spike can support several competing hypotheses. The analysis highlights five recurring risks: scalable identity abuse and account takeover, payment and ledger exploitation, replay or duplication of canonical events, latency-sensitive denial of service, and misuse of privileged detection or enforcement tools. The paper concludes that monetized cheering should be protected as an end-to-end pipeline, with safeguards placed at the relevant trust boundaries: server-side event generation, idempotency and replay resistance, adaptive throttling, graceful degradation, tamper-evident audit trails, privacy-aware measurement, and least-privilege operational access.**

***Keywords*—threat modelling; Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE); Damage, Reproducibility, Exploitability, Affected Users, and Discoverability (DREAD); live streaming security; monetization fraud; anomaly detection; trust and safety**

I. Introduction

Live-streaming platforms such as Twitch and YouTube turn paid interactions into public signals: cheers, highlighted messages, gifts, badges, and overlay alerts. These signals carry status for viewers and revenue for creators, so they influence participation, community norms, and platform economics [1–3].

The focus of this paper is anomalous cheering: episodes that depart sharply from what would normally be expected for a channel or platform after account history, device and network reputation, timing, and stream context are taken into account. A spike may be harmless, such as a genuine one-off donation, but the same pattern can also reflect automated amplification, payment fraud, chargeback abuse, or a coordinated attempt to distort revenue and visibility.

Recent work on suspicious cheering with Bits shows that earnings can jump in ways that ordinary channel dynamics do not easily explain and that may warrant closer review [4]. Research on coordinated abuse in live-streaming communities, including hate raids, also shows that real-time interaction surfaces can be exploited at scale [5]. Anomalous cheering is therefore not merely an analytics problem; it also concerns security, governance, and the everyday experience of creators and viewers.

Anomaly detection can flag unusual episodes, but an alert alone rarely explains what has happened, who may be harmed, or which part of the platform should be addressed first. Threat modelling helps close that gap by connecting observed patterns to system design, security goals, and concrete controls. Although previous work has described irregular cheering and proposed detection approaches, less work follows cheering end-to-end through identity, payments, ledgers, and internal detection and enforcement tools using a Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) model.

Monetized cheering is not simply a payment flow. It links public social signals, such as badges, alerts, and ranking effects, to real value transfer through wallets, ledgers, settlement, and payouts. When cheering departs from a channel's normal pattern, whether through high value, unusual volume, bursty timing, or suspicious attribution, the consequences can affect income, visibility, and the tone of the live interaction.

A single visible spike can arise from several technical and operational causes: identity-driven automation or account compromise; payment and chargeback abuse;

Manuscript received March 2, 2026; revised March 25, 2026; accepted May 18, 2026; published July 23, 2026.

replay or duplication in event processing; denial-of-service pressure on latency-sensitive components; or misuse of privileged access in internal tools. Because anomaly alerts rarely identify intent, affected assets, or the trust boundaries involved, their operational value can be limited. This paper translates anomalous cheering into an architecture-grounded threat model that clarifies what is at risk, what adversaries can plausibly do, and which controls are likely to matter most.

The paper asks: How can a STRIDE-based model of monetized live-stream cheering identify key assets and trust boundaries, prioritize threats with a simple Damage, Reproducibility, Exploitability, Affected Users, and Discoverability (DREAD) scale, connect anomalies to practical mitigations, and clarify trade-offs around human review, privacy, and oversight?

To address this question, the paper develops three outputs: (1) a reusable reference architecture for monetized cheering, including detection and operational response, (2) a STRIDE per-element threat register that links scenarios to assets and controls, and (3) a simple DREAD-style prioritization with a worked example showing how one anomalous episode can support competing threat hypotheses. The model is conceptual and is drawn from common platform designs and published research, not from proprietary platform data or internal incident reports.

Existing research on anomaly detection, fraud analytics, and coordinated abuse often treats these problems separately. It rarely models the chain linking identity, payments, event processing, and real-time social signaling as a security pipeline. The result is a gap between seeing anomalous behavior and understanding how it maps to system-level vulnerabilities, trust-boundary crossings, and controls. The model proposed here addresses that gap for gaming and live-streaming platforms.

The contribution is threefold. First, the paper introduces a reference model for monetized cheering that treats identity, payments, wallets and ledgers, event streaming, anomaly detection, and trust and safety operations as one pipeline with explicit trust boundaries and handoff risks. Second, it provides a STRIDE per-element threat register that connects threat scenarios to affected assets, trust-boundary crossings, and practical controls, giving practitioners a repeatable basis for platform-specific threat modelling. Third, it links the threat register to an ordinal DREAD prioritization, a worked multi-hypothesis anomaly example, and a privacy-security-fairness trade-off analysis. Together, these outputs show how anomaly findings can be translated into measured operational decisions.

The rest of the paper proceeds as follows: Section II reviews related work; Section III describes the methodology; Section IV presents the threat map, prioritization, worked example, and trade-off analysis; Section V discusses implications for platform design, the privacy-security-fairness trade-off, and cross-platform practice; and Section VI concludes.

II. Literature Review and Related Work

Research relevant to anomalous cheering spans live-streaming participation, inauthentic engagement, coordinated abuse, fraud analytics, anomaly detection, and threat modelling. Across these areas, cheering appears less as a simple engagement metric than as a pipeline where social interaction and value transfer meet.

A. Cheering as Social and Technical Monetisation

Cheering and related monetization mechanisms operate as both payment-adjacent infrastructure and social signaling systems. Hamilton *et al.* [1] describe participatory communities around Twitch's live mixed-media environment, and Hilvert-Bruce *et al.* [2] identify viewer motivations such as community, entertainment, and recognition. Johnson and Woodcock [3] analyze Twitch monetization and show how cheering sits within revenue strategies shaped by platform features and cultural norms. Manipulating cheering can produce financial gain alongside social effects such as visibility, status, and attention.

B. Automation, Fake Engagement, and Coordinated Abuse

Automation and inauthentic engagement are well documented across online ecosystems. Ferrara *et al.* [6] surveyed social bots and their ability to imitate user behavior and distort engagement signals. Nevado-Catalán *et al.* [7] show that fake social-media engagement is commercially packaged and configurable. In live streaming, Cai *et al.* [5] examine coordinated hate raids that combine human and automated behavior. These findings matter for cheering because the same economics of automation and coordination can be redirected towards monetized interaction.

C. Fraud Analytics and Anomaly Detection

Fraud detection research points to two constraints that matter here: attackers adapt, and labelled data are often scarce. Anomaly detection is useful in this setting, particularly when it is paired with human review. Bolton and Hand [8] review statistical fraud detection and stress operational constraints and cost trade-offs. Chandola *et al.* [9] survey anomaly detection methods and note persistent challenges around interpretability and false positives. Bhattacharyya *et al.* [10] compare data-mining approaches for credit card fraud and emphasize feature engineering and evaluation. Ding *et al.* [11] show how anomaly-detection output can be connected to explanation and action. In the cheering domain, Eminov *et al.* [4] show that statistical signals can surface Bits-related revenue episodes that warrant human verification.

D. Threat Modelling with STRIDE, DREAD, and Privacy Threat Models

Threat modelling links system design to attacker behavior in a concrete, reviewable form. STRIDE is widely used to categorize threats and is often applied per element of a Data-Flow Diagram (DFD) to reduce omissions [12]. DREAD can help triage a long threat register, but its numeric totals should be treated as ordinal

signals rather than precise measures. Monetized cheering also raises privacy risks because logs, measurement data, and model features can expose linkable behavioral traces. Deng *et al.* [13] propose LINDDUN, a privacy threat-modelling methodology that can strengthen analysis of information disclosure, linkability, and inference risks when measurement data is central.

To assess novelty, we searched ACM Digital Library, IEEE Xplore, and Google Scholar using live-streaming and monetization terms (cheering, Bits, gifts) together with threat-modelling terms (STRIDE, DREAD, threat model). This search found work on cheering anomalies and coordinated abuse, but not a peer-reviewed, end-to-end STRIDE per-element threat model of a monetized cheering pipeline that treats payment and ledger flows and the detection-and-enforcement toolchain as first-class components. The approach proposed here is not a detection algorithm. It is a conceptual and architectural model that complements detection by linking observed anomalies to system-level threats, trust boundaries, and mitigation options in monetized live streaming.

E. Availability Threats and DDoS Context

Cheering endpoints must handle legitimate bursts, such as raids and major events. That same tolerance makes them attractive targets for denial-of-service attacks that exploit volume, amplification, or hot-spot effects in real-time aggregation. Mirkovic and Reiher [14] provide a Distributed Denial-of-Service (DDoS) taxonomy that remains useful for structuring availability threats and layered defenses in latency-sensitive systems.

F. Value Transfer Structures and Money-Laundering Risk

When cheering uses virtual currency and payout logic, the platform mediates a form of value transfer. Levi and Reuter [15] describe how money laundering exploits systems that permit layered transfers and weak attribution, while the Financial Action Task Force [16] outlines Anti-Money Laundering and Counter-Financing of Terrorism (AML/CFT) risks associated with virtual currencies. Legal obligations vary by jurisdiction and product design, but the structural issue is relevant for security analysis: anomalous cheering may indicate fraud or laundering-like value routing, not merely unusual engagement.

III. Methodology

The analysis proceeds in three steps. First, it defines a reference cheering pipeline and its trust boundaries. Second, it enumerates threats with STRIDE per element. Third, it uses a simple ordinal DREAD rubric to prioritize scenarios that require mitigation planning.

The study does not rely on proprietary platform data, internal incident reports, or workshop artefacts. Instead, it builds a reference architecture from common streaming designs and published research on live streaming, automation, and fraud [4–6]. The result should be read as a reusable model that platforms can adapt and validate empirically.

A. Definition of Anomalous Cheering

Anomalous cheering is defined here as cheering behavior that deviates sharply from expected baselines in ways that may indicate adversarial behavior or control failures. The deviation may be in volume, total value, timing, attribution, or consistency checks between canonical event logs and aggregated metrics. Because some anomalies are benign, the threat model does not assume that every outlier is malicious. Its purpose is to explain how similar signatures can arise from different threats and to guide proportionate responses.

B. Reference Architecture and Trust Boundaries

Fig. 1 presents a simplified DFD for a monetized cheering pipeline and shows how digital donations, including Twitch Bits, pass through live-streaming systems. The model distinguishes three trust zones: the Client Zone (Untrusted), the Platform Internal Network (Trusted-by-Design), and the Operations/Analytics Layer (Separate Internal Trust Zone). These zones differ in platform control, identity assurance, and exposure to attack. At each stage, user-initiated donations, including those from spoofed, anonymized, or automated clients, may become both monetary and social effects, so integrity across boundaries is critical. The main trust boundaries are the client-edge interface, the platform-payment processor interface, and the production-to-operations interface [12].

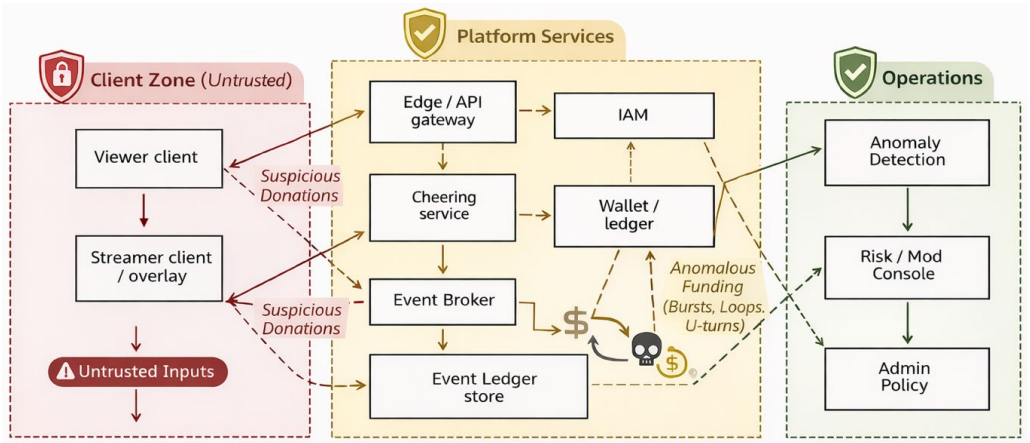

Fig. 1. Data flow diagram of a monetized cheering pipeline with primary trust zones.

External/Public Internet (untrusted): This zone covers traffic outside the platform-controlled perimeter, including viewer clients, streamer overlays, and their networks. It is the most exposed entry point for threat actors using botnet relays, tampered proxies, or repeat-offender accounts.

Platform Internal Network (trusted-by-design): This zone contains the platform logic and processing layer, including the Edge/API Gateway, IAM, Cheering Service, Wallet/Ledger, Event Broker, and Event Ledger Store. These services are expected to use mutual authentication, encryption, and intra-service access policies while validating donations, sequencing events, and processing financial state transitions.

Payment processing services (external trust zone): This zone includes third-party services, such as Stripe or PayPal, that authenticate and capture real-world payments. They can validate payment instruments, but the platform still must decide whether the resulting in-platform event, overlay, or downstream payout is legitimate. A shared-responsibility model is needed because a successfully authorized payment does not by itself prove legitimate downstream intent or behavior.

Operations/Analytics (separate internal trust zone): This zone includes platform-side security and moderation services, such as Anomaly Detection, Risk/Moderator Consoles, and Admin Policy interfaces. These services should be separated from production infrastructure and protected with stronger access control [17], audit policy, and model-deployment controls because they underpin human-in-the-loop and automated enforcement.

Key trust boundaries (illustrative) are:

- TB1 (Client/Edge): External/Public Internet to Edge/API Gateway. This boundary applies request authentication, anti-automation controls, and schema validation. It is where malformed traffic, bot-based cheering, and basic evasion attempts should be stopped.
- TB2 (Payments): Platform Internal Network to Third-Party Payment Processor. Payments may be authenticated externally, but the conversion of an approved payment into an in-platform cheer event is controlled by platform logic. This boundary is sensitive because laundering-like behavior can exploit race conditions, refund cycles, or false event triggers.
- TB3 (Prod/Ops): Production Services and Data Stores to Operations/Analytics Tooling. This boundary controls the movement of telemetry, transaction logs, and event data into detection systems. It protects model-training data and helps ensure that moderation decisions use complete and reliable inputs.
- TB4 (Privileged Change Paths): Admin Console/Policy Service to Configuration Store and Deployment Pipelines. Changes to detection thresholds, risk classifiers, or streamer moderation rules cross this high-risk boundary. Role separation, review gates, and tamper-evident logging should be required.

C. Payment Processor Boundary and Shared Responsibility Model

Cheering pipelines typically integrate an external payment gateway (e.g., Stripe) or platform-managed stored value. In a shared-responsibility model, the gateway handles payment instruments and transaction authorization or capture, while the platform controls the business logic that turns an authorization into (a) a canonical cheer event, (b) a wallet/ledger mutation, and (c) a public social signal such as an overlay alert, badge, or ranking effect. A secure gateway does not prevent abuses that exploit platform-side sequencing, idempotency, reconciliation, or UI emission rules, such as rendering a visible cheer before settlement or allowing replayed intents to create duplicate social signals.

We treat this interface as a first-class trust boundary in the STRIDE per-element analysis. Threats are enumerated not only for payment processing, but also for the platform-controlled post-payment pipeline that generates the social signal and reflects settlements, refunds, and chargebacks in ledgers and downstream aggregates.

D. Assets and Attacker Model

The assets considered in this analysis are: (1) integrity and attribution of canonical cheer events; (2) correct wallet and ledger accounting, settlement, and reconciliation; (3) availability and latency of real-time interaction and overlay updates; (4) privacy of viewer-linked transaction and behavioral measurement data; (5) integrity and provenance of anomaly scores, model configurations, and enforcement actions; and (6) auditability for dispute handling and oversight.

The attacker model includes four classes: external attackers without legitimate accounts; malicious or compromised viewers who can initiate cheers; complicit streamers who coordinate manipulation; and insiders, or compromised internal accounts, with access to operational consoles or configuration pipelines. The model assumes that attackers can automate client behavior at scale [6] and that manipulation can be commercialized [7].

E. STRIDE per Element Procedure

STRIDE is applied per element [12] by decomposing the reference system into DFD elements: external entities, processes, data stores, and data flows. For each element and each trust-boundary crossing, we consider the STRIDE categories, write threat statements in a consistent actor-action-asset-impact form, and attach candidate controls at the stage where they apply.

Operational enumeration steps are:

(1) Define scope and assets: specify what constitutes a canonical cheer event and which properties must be protected (integrity, attribution, ordering, settlement linkage).
(2) Draw or refine the DFD: list entities, processes, data stores, and flows; mark trust zones and boundary crossings (TB1-TB4).
(3) List threats per element: for each element and boundary crossing, iterate through STRIDE categories and write threat statements using the same structure and level of specificity.

(4) Attach controls: map mitigations to the stage where they apply (edge, service-to-service, data store, operations tooling) and label whether they are preventive, detective, or responsive.
(5) Consolidate and de-duplicate: merge equivalent threats that differ only in minor preconditions; retain variants when mitigations or affected assets differ materially.
(6) Review for completeness: verify that each trust boundary has plausible spoofing, tampering, repudiation, information disclosure, availability, and privilege-abuse paths considered, including production-to-operations and privileged change channels.

F. DREAD-Style Prioritisation using a Simple Scale

STRIDE enumeration usually produces more scenarios than a team can address at once, so a second triage step is needed. We use a DREAD-style rubric (damage potential, reproducibility, exploitability, affected users, and discoverability) to prioritize threats. Other approaches were considered. Process for Attack Simulation and Threat Analysis (PASTA) is useful for attacker-centered risk analysis but requires more organization-specific business and operational evidence than a conceptual model can assume. Attack trees support detailed reasoning about individual attack paths, but they become unwieldy across a multi-component pipeline with many trust boundaries. Visual, Agile, and Simple Threat (VAST) is designed for large agile delivery environments, while Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) is better aligned with enterprise-level risk governance than with pipeline-level threat enumeration. STRIDE per element fits the present task because it maps directly onto a DFD and supports systematic coverage of components and boundary crossings. DREAD is used only as a lightweight triage tool: it does not claim probabilistic precision, but it gives enough structure to rank threats for response planning.

To avoid spurious precision, each dimension uses a 1–3 scale and totals are interpreted qualitatively. Where feasible, teams should reconcile scores across more than one analyst to reduce subjectivity [8].

The ordinal anchors used for the DREAD scoring are:

Damage potential: 1 = localized/minor (low financial or operational impact); 2 = moderate (meaningful financial loss or repeated user harm); 3 = high (systemic financial, availability, regulatory, or reputational impact).

Reproducibility: 1 = difficult to repeat (fragile conditions); 2 = repeatable with effort; 3 = easy to repeat once set up (automation-friendly).

Exploitability: 1 = requires specialized access or rare preconditions; 2 = requires moderate skill/resources; 3 = low barrier and can be scaled with common tooling.

Affected users: 1 = few users/channels; 2 = many users or a significant subset; 3 = platform-wide or highly visible multi-channel impact.

Discoverability: 1 = hard for attackers to identify/control path; 2 = discoverable with experimentation; 3 = obvious from interfaces or common knowledge in attacker markets.

DREAD-style scores depend on the attacker being assumed. The highest-priority spoofing scenario assumes a well-resourced fraud actor with access to residential proxy networks, automated account tooling, and pools of stolen or synthetic identities. Less capable attackers would usually score lower on reproducibility, exploitability, and affected users, which may change the ordering of mid-tier risks.

Even on a simple scale, summed dimensions can imply more numeric certainty than the evidence supports. Totals are used only for coarse triage, not for interpreting one-point differences. Platforms should re-score under at least two attacker personas, treat one-point gaps as non-actionable unless supported by qualitative evidence, and reconcile scores across multiple analysts where possible.

G. Outputs of the Analysis

The analysis produces five outputs: a simplified DFD, an expanded STRIDE threat map, an ordinal DREAD-style prioritization, a privacy and oversight trade-off table, and a worked example that maps one anomalous episode to competing threat hypotheses and responses.

H. Additional System Decomposition (DFD Elements)

For the STRIDE per-element analysis, the DFD is decomposed into external entities (viewer client, streamer client or overlay, payment processor), processes (edge/API gateway, IAM, cheering service, wallet/ledger, event broker, real-time aggregator, anomaly detection, risk and moderation console, admin policy service), data stores (event log, ledger store, configuration store), and key data flows (cheer request, authentication check, wallet debit or credit, canonical event publication, aggregate update, overlay update, measurement data export to detection, console alerting, and administrative configuration changes). The main trust boundaries are the client-edge interface, the platform-payment interface, and the production-operations interface.

These boundaries also indicate where controls belong. Rate limiting, coarse device fingerprints or device-bound attestations, and schema validation mainly apply at TB1 (Client/Edge). Payment validation and sequencing protection apply at TB2 (Payments). Audit logging, model governance, access control, and privileged change controls apply at TB3 and TB4, where operational tooling and privileged change paths introduce additional risk. This mapping helps keep mitigation decisions tied to architecture rather than to generic control lists.

IV. Results

The results are organized around four outputs from the threat-modelling exercise: (i) an expanded STRIDE per-element threat map for the reference cheering pipeline, (ii) a simple DREAD-style prioritization to support triage, (iii) a worked example that translates an anomalous episode into explicit threat hypotheses and controls, and (iv) a summary of privacy, security, and oversight trade-offs for representative mitigations.

A. STRIDE Threat Map

Table I summarizes why STRIDE per element, supplemented by DREAD, was selected for this reference model. Table II then summarizes representative STRIDE threats and links each scenario to assets at risk and plausible controls. The register is not exhaustive. It is intended as a starting point that should be applied to a platform-specific DFD and extended when implementation details introduce additional attack paths.

TABLE I. COMPARISON OF THREAT MODELLING APPROACHES ACROSS CAPABILITIES RELEVANT TO MONETIZED LIVE-STREAMING SYSTEMS

Approach	End-to-End Pipeline Coverage	Models Payments and Social Signals	Trust Boundary Mapping	Threat Prioritisation	Domain-Specific to Live-Streaming Monetisation
STRIDE (generic)	No	No	Partial	No	No
Fraud detection methods	Partial	Partial	No	No	No
Abuse/coordinated attack detection	Partial	No	No	No	No
Anomaly detection methods	No	No	No	No	No
Proposed model	Yes	Yes	Yes	Yes	Yes

TABLE II. EXPANDED STRIDE THREAT MAP FOR ANOMALOUS CHEERING WITH CONTROL TYPES

Stride	Threat Scenario (Anomalous Cheering)	Primary Assets at Risk	Key Controls (Technical Specificity)	Control Type(s)
Spoofing	Automated or compromised accounts create synchronised cheering bursts that appear to come from many legitimate viewers.	Identity integrity; fair monetisation and ranking; reliable anomaly signals.	Risk-based step-up authentication; privacy-limited device fingerprints or device-bound attestations; proxy/IP/geo/ Autonomous System Number (ASN) velocity checks; per-identity, device, and channel rate limits; account-creation friction and bot detection [6].	P/D
Tampering	Replay, duplicate processing, altered labels or thresholds, or unauthorised ledger changes inflate or hide cheer events.	Canonical event integrity; ledger correctness; auditability; trust in detection outputs.	Server-side event generation; idempotency at write and aggregation stages; event-ledger reconciliation; append-only ledger semantics; signed, change-controlled model/rule deployment with rollback [12].	P/D/R
Repudiation	Viewers or streamers deny responsibility for anomalous episodes during disputes, enforcement, account-takeover claims, or chargebacks.	Non-repudiation evidence; chargeback handling; enforcement legitimacy.	Append-only audit trails with stable identifiers; signed high-value cheer intents through step-up authentication; trusted time-stamping [18]; retention aligned to dispute windows; risk-based holds or escrow [8].	D/R
Information disclosure	Logs, dashboards, analytics exports, or leaked model details expose sensitive measurement data and help attackers evade controls.	User privacy; financial confidentiality; robustness of detection and enforcement.	Least-privilege dashboard/log access; encryption; minimised or aggregated features; redaction or tokenisation for non-security roles; export controls, Data Loss Prevention (DLP) monitoring, and LINDDUN-informed privacy review [13].	P/D
Denial of service	Flooded cheering endpoints, overlay updates, or hot-key patterns degrade latency, stream quality, and aggregation stability.	Availability; real-time user experience; operational stability during major events.	Adaptive throttling with backpressure; cost-based limits; per-channel overlay coalescing; guarded autoscaling; graceful degradation of non-critical effects; layered DDoS defences [14].	P/R
Elevation of privilege	Internal-tool flaws or misconfigurations let attackers or insiders alter thresholds, whitelist abusive actors, or disable controls.	Enforcement integrity; financial integrity; compliance posture; trust in oversight.	Role-Based Access Control (RBAC), Multi-Factor Authentication (MFA), and just-in-time privilege elevation; dual authorisation for whitelisting, fraud thresholds, and model deployment; signed change records; immutable audit trails; access reviews [17].	P/D/R
Spoofing	Forged or replayed server-to-overlay messages display a cheer without a matching canonical event.	Integrity of public social signals; streamer trust; incident-response evidence.	Signed overlay or webhook payloads; mutual TLS (mTLS) or equivalent origin verification; nonces and replay windows; key rotation; monitoring for overlay-log mismatches.	P/D
Tampering	Payment sequencing (TOCTOU) abuse renders a visible cheer before settlement, followed by refund or chargeback abuse.	Monetisation and ranking integrity; ledger correctness; creator fairness; chargeback exposure.	Two-phase event state (pending to settled); escrow for high-risk cases; payout credit after settlement; automated chargeback/refund reversal; continuous event-ledger reconciliation.	P/D/R
Tampering	Detection evasion or poisoning uses repeated small anomalies and unsafe labels to normalise botnet behaviour.	Anomaly-output integrity; enforcement legitimacy; long-term control resilience.	Keep investigator labels separate from online learning; quarantine overrides; require dual review for baseline-changing labels; monitor drift and poisoning; protect validation sets.	P/D/R
Repudiation	A privileged operator or compromised admin account changes fraud thresholds or whitelists and later denies responsibility.	Oversight and auditability; enforcement integrity; post-incident forensics.	Signed admin change requests; immutable audit logs with trusted time-stamping [18]; dual authorisation; periodic attestation and review of configuration diffs.	P/D
Information disclosure	Public APIs expose internal or sequential identifiers that help attackers map enforcement states or infer thresholds.	Privacy and confidentiality; robustness of fraud controls; attacker reconnaissance capability.	Opaque, non-sequential IDs; strict per-object authorisation; minimised responses; anti-scraping throttles; enumeration monitoring [19, 20].	P/D
Denial of service	Thousands of micro-cheers overwhelm the aggregator or overlay pipeline through per-event processing and fan-out.	Real-time pipeline availability and latency; stream quality; operational stability.	Micro-batching and per-channel coalescing; capped overlay frequency; ingestion pre-aggregation; approximate counters or rolling windows; cost-based CPU/database throttling.	P/R

Note: Control type(s): P = Prevent; D = Detect; R = Respond.

B. DREAD-Style Prioritisation (Illustrative)

The STRIDE map in Table II shows three main patterns. First, the most important attacks often cross trust boundaries: scalable identity spoofing at TB1 can enable payment abuse at TB2 and may remain hidden if operational tooling at TB3 or TB4 is compromised. Second, tampering threats can be hard to detect when baselines are poisoned or replayed events are made to resemble legitimate canonical events; in some cases, the problem becomes visible only during a dispute or audit. Third, denial-of-service in cheering is not merely a generic availability issue. Because overlays are real-time social signals, even short disruptions can damage the creator-viewer experience and the perceived reliability of monetization. These observations inform the prioritization in Table III.

Table III prioritizes a subset of high-impact threats. The 1–3 DREAD scores are ordinal and are used for coarse ordering, not precise quantification. The ordering reflects three practical concerns: automation scales [6], fraud and chargeback dynamics create direct financial exposure [8, 10], and real-time systems are sensitive to bursty overload [14].

TABLE III. TOP PRIORITIZED THREATS USING ORDINAL DREAD-STYLE SCORING (D/R/E/A/DI)

ID	Threat Summary	DREAD Scores (D/R/E/A/Di)	Total	Rationale
P1	Scalable spoofing via synthetic/compromised identities produces inauthentic cheering bursts and manipulates monetisation or visibility signals.	3/3/3/3/3	15	Assumes a well-resourced fraud actor using residential proxies and pools of stolen or synthetic identities; automation makes spoofing scalable and repeatable, with platform-wide visibility impact.
P2	Payment and ledger abuse: stolen instruments, refund/chargeback patterns, and laundering-like routing through cheering and payouts.	3/3/2/3/2	13	Fraud and chargebacks create direct financial exposure; the attack is reproducible once a workflow is established, although gateway risk controls and step-up checks may raise friction.
P3	Flooding of cheering endpoints or overlay amplification degrades availability/latency during high-visibility events.	3/3/2/3/2	13	Real-time overlays and low-latency pipelines have little tolerance for bursts; even short spikes can degrade stream quality, making damage and affected-users high. Exploitability is moderate because platforms can throttle, but attackers can distribute load.
P4	Replay/duplication or idempotency failure inflates canonical events and aggregates without breaking cryptography.	2/3/3/3/2	13	At-least-once messaging, retries, and client replays make duplication a realistic integrity risk; once an idempotency gap is found, replay is automation-friendly and impacts many users via inflated aggregates and payouts.
P5	Elevation of privilege in internal consoles or model deployment pipelines suppresses detection or alters enforcement outcomes.	3/2/2/3/2	12	Usually requires privileged access, such as an insider, compromised admin account, or deployment-path flaw; impact remains high because threshold changes or whitelisting can suppress detection at scale.
P6	Disclosure of viewer-linked transaction histories or model features enables profiling and evasion of fraud controls.	3/2/2/3/2	12	Leakage of viewer-linked measurement data or model features creates privacy/regulatory harm and enables evasion; exploitation is moderate (often via over-broad dashboard/log access or API leakage) but can affect many users once exposed.

Network failures and retries under at-least-once delivery can cause the same cheer event to be processed more than once. For that reason, idempotency must be enforced at more than one layer. At the API gateway, idempotency keys can suppress repeated client requests within a bounded window. The wallet and ledger also need idempotent state transitions, with each transaction tied to a unique immutable identifier so that duplicate writes are rejected or safely coalesced. This layered design prevents duplicated events from producing double-spending or inconsistent ledger state.

Three points stand out from Table III. First, P1 (scalable spoofing) and P2 (payment and ledger abuse) combine high damage with high reproducibility, so identity abuse and payment sequencing should be early priorities for platforms introducing monetized cheering. Second, P3 (DoS) and P4 (replay/duplication) share a structural weakness in at-least-once delivery and retry behavior; idempotency enforcement and overlay coalescing can therefore reduce exposure to both. Third, P5 (privilege escalation) and P6 (information disclosure) score slightly lower because they often require insider access or excessive log access, but they still require governance and access-control attention because they threaten detection integrity.

C. Interpreting Anomalies as Threat Hypotheses

Threat modelling is useful here because an observed anomaly can support several competing hypotheses, rather than a single fraud label. A sudden spike in cheering value, for example, might reflect identity-driven automation, payment fraud that later appears as chargebacks, replay or duplication from idempotency failures, a coordinated harassment campaign, or a benign attention shock. Disambiguation depends on context: account history, payment signals, device or network clustering, temporal patterns, and consistency checks between canonical events and aggregates [9].

Anomaly detection should be part of a security feedback loop rather than a descriptive dashboard. Human review can connect an alert to a response, such as placing funds on hold, triggering step-up verification, throttling effects, or routing a case for investigation. Audit records are essential because actions must remain accountable and fair [11]. The worked example below shows how this mapping can be made explicit.

Human review improves interpretability and accountability, but it will not scale on its own during platform-wide bursts such as Scenario P1. Platforms can use intermediate automated controls to reduce the load on

investigators. Step-up authentication or device revalidation can be triggered for high-risk transactions, while escrow or delayed settlement can limit financial exposure without interrupting the on-stream experience. These measures let the system defer final decisions and reserve human review for ambiguous or high-impact cases.

D. Worked Example Threat Scenario

Scenario: A coordinated campaign produces a sudden burst of high-value cheering across several small channels. The burst is driven by automated identities and is followed by disputes or chargebacks. The surface signal could still be benign, but the scenario illustrates how one observation can be translated into explicit threat hypotheses and proportionate controls.

1) Attacker's goal and plausible motivation

A plausible attacker goal is to move value through the platform or manipulate perceived popularity. Markets for fake engagement show that attention manipulation can be bought and scaled [7], and fraud analytics research shows that attackers target monetizable signals and adapt to controls [8]. In monetized cheering systems, these incentives can overlap: automated identities generate visible cheers while payment abuse is attempted through refunds and chargebacks.

2) Preconditions (non-exhaustive)

The campaign assumes multiple viewer identities, the ability to initiate cheering transactions at scale, and some way of funding those transactions. The identities may be synthetic, compromised, or rented; the funding may be legitimate or fraudulent. The analysis does not assume a break in cryptography. It assumes abuse at the level of automation, orchestration, and policy friction, consistent with social-bot research [6].

3) Observable anomaly signals and competing hypotheses

From an anomaly-detection perspective, the main symptom is a sharp deviation from baseline cheering value or count [9]. Additional signals help separate competing explanations. New-account clustering, shared device or network infrastructure, short interaction histories, and synchronized timing support a spoofing or automation hypothesis. Duplicated idempotency keys or event identifiers support a replay or duplication hypothesis. A spike followed by disputes or chargebacks supports a payment-fraud hypothesis. Because anomalies can be benign, investigators should also consider raids, charity drives, creator milestones, and other contextual explanations [4].

4) STRIDE mapping and example controls

Spoofing is relevant when synthetic or compromised identities are used to simulate broad participation. Tampering is relevant when replay or duplication inflates canonical events, or when labels and thresholds are changed to suppress flags. Repudiation matters during disputes, such as account-takeover claims, and depends on strong audit trails. Denial of service may co-occur when the same automation floods interaction endpoints. Elevation of privilege becomes critical if attackers or insiders can change enforcement policies or model configurations.

Controls should be layered and proportionate: step-up verification or velocity checks for unusually high-value cheers; temporary holds or delayed settlement for high-risk transactions; per-identity and per-channel quotas during bursts; and hardened operational tooling with separation of duties and tamper-evident logging for policy and model changes [17]. Human-in-the-loop anomaly management helps connect detection to explanation and action [11].

5) Illustrative DREAD scoring

Under the ordinal rubric, this scenario scores high on damage, reproducibility, and affected users because it can combine financial loss, trust erosion, automation, and impact across several channels. Exploitability and discoverability depend on platform friction. A representative score is D/R/E/A/Di = 3/3/2/3/2 (total 13), consistent with Table III.

E. Privacy and Oversight Trade-Offs

Fraud and abuse controls in monetized cheering pipelines can protect the platform while also affecting privacy, creator fairness, and stream continuity. Table IV summarizes representative trade-offs and safeguards, with particular attention to limiting false-positive harm while preserving investigative value.

Regulation also shapes how privacy requirements and security controls are balanced. Device fingerprinting or continuous behavior monitoring may require stronger transparency, data minimization, or consent mechanisms in some jurisdictions. Platforms can respond with coarser signals, shorter retention periods, and risk-based activation, even if these choices reduce detection effectiveness. Other jurisdictions may permit broader use of such signals. Controls should be adaptable to regulatory context rather than applied globally without adjustment.

V. Discussion

The discussion draws three implications from the threat register: how platforms should design and deploy protections, how they should manage trade-offs among privacy, security, and fairness, and how these lessons generalize across other monetized interaction systems.

A. Implications for Platform Design and Protection Deployment

The threat map suggests that cheering should be defended as a whole monetization pipeline, not as a single interface feature. The most important controls sit at architectural choke points: server-side event generation; wallet and ledger correctness with reconciliation; adaptive throttling and graceful degradation for latency-sensitive components; and hardened operational tooling. These priorities are consistent with fraud analytics, where detection must be operationalized, cost-aware, and resilient to adversarial adaptation [8, 10].

TABLE IV. PRIVACY AND OVERSIGHT TRADE-OFFS FOR REPRESENTATIVE MITIGATIONS (ILLUSTRATIVE)

Control/Practice	Security Benefit	Privacy Risk	Oversight/Fairness Risk	Design Mitigation/Safeguard
Device fingerprinting/ device-bound attestations	Improves bot and multi-account detection for spoofing at scale.	Increases linkability across sessions and can become a high-sensitivity identifier.	May disproportionately flag shared devices, cybercafes, or low-resource users.	Minimise retention; use coarse-grained features; provide appeals; restrict access; document purpose and limits.
Soft verification with escrow holds (e.g., 72-h hold for high-risk cheers)	Reduces fraud loss while preserving on-stream continuity.	Requires holding transaction metadata longer and may expand investigative logging.	Delayed payouts can disadvantage small creators and create perceived unfairness.	Use risk-based thresholds; make holds transparent; time-box holds; allow rapid appeal and manual release for trusted channels.
Computational cost throttling and micro-batching	Protects real-time systems from DoS and algorithmic complexity attacks.	Generally low privacy impact (uses system metrics rather than user identity).	Aggressive throttling can mute legitimate high-energy moments during large events.	Prefer graceful degradation (coalesce updates) over hard blocks; publish rate-limit semantics; monitor for disparate impact.
Human review labelling ("mark as safe")	Provides interpretability and operational control for ambiguous anomalies.	Labels can reveal sensitive behavioural inferences if broadly accessible.	Subjective judgements can introduce bias; labels can be gamed (poisoning).	Separate duties; require dual review for baseline-shifting labels; maintain audit trails; keep training-data oversight separate from enforcement.
API metadata minimisation (opaque IDs, minimised responses)	Reduces reconnaissance and evasion by limiting attacker visibility into internals.	Decreases inadvertent exposure of internal identifiers and linkage data.	Can reduce transparency for creators debugging legitimate issues.	Provide authenticated troubleshooting views; keep public APIs minimal; document stable external identifiers.
Four-eyes principle for admin actions	Mitigates insider risk and prevents unilateral threshold/whitelist changes.	Low privacy impact (oversight control).	Adds latency to urgent responses; may slow Trust and Safety (T&S) operations.	Support break-glass with post-hoc review; require signed approvals; implement just-in-time access and periodic access reviews.

B. The Trade-Off between Privacy, Security, and Fairness

Aggressive fraud controls can harm legitimate viewers and creators when benign outliers resemble adversarial bursts. This risk is especially acute for small channels, where one legitimate high-value donation can dominate historical baselines [4]. Platforms need proportionate responses, transparent appeal routes, and consistent evidentiary standards. Measurement data used for detection can also introduce privacy risk: linking transaction histories to device and network signals can expose sensitive behavioral traces. LINDDUN offers a structured way to surface these privacy threats and map safeguards such as minimization, aggregation, and access control [13].

False positives are not evenly distributed. For small or newly monetizing channels, a single genuine high-value donation can depart sharply from historical baselines, increasing the likelihood of an anomalous flag even when no abuse is present. Because smaller channels often include new or marginalized creators, simple thresholds can create a social and technical bias: delayed payouts, uncertainty, and lower trust in oversight. Review queues need to account for channel maturity, expected variance, and context signals, such as raids or charity drives, when available.

Fraud controls should also preserve the flow of the stream where the risk does not justify immediate blocking. One option is soft verification: the cheer remains visible in real time, but funds are held in escrow, for example for 72 h, before being credited to the streamer payout ledger if the risk score exceeds a threshold. This replaces a binary allow-or-deny decision with a more proportionate control loop. It can limit chargeback exposure and give investigators time to validate outliers without publicly undermining legitimate donors or creators.

C. Cross-Platform General Practice Recommendations

Across the cheering pipeline, several practical safeguards are useful regardless of the specific threat. First, platforms need reliable visibility. The edge, wallet and ledger, event broker, real-time aggregator, and overlays should be instrumented so that volume, value, latency, and error rates can be analyzed together. When an alert fires, investigators should be able to trace a cheer from request to settlement using consistent identifiers and reconcile the on-screen signal with the canonical event log and ledger.

Second, operational change should be treated as a security boundary. Fraud rules, model updates, and configuration changes should go through review, signing, and rollback in the same disciplined way as production code, with tamper-evident audit records and separation of duties for high-impact actions. These practices should be paired with service identity and key hygiene, including mutual TLS (mTLS), short-lived credentials, and routine rotation; privacy-aware telemetry that limits collection, retention, and access; and rehearsed incident playbooks for throttling or graceful degradation during surges. Baseline API and application security controls [19, 20] also help turn common failure modes into routine engineering checks.

The same logic applies beyond live streaming to systems that combine real-time interaction, financial transactions, and user-facing signals, including online marketplaces, gaming ecosystems, and monetized social media platforms. In these settings, mapping anomalous behavior to system-level threats and operational controls can support more consistent decisions under high-volume or adversarial conditions. Security, fraud, and operations should be treated as architectural concerns rather than as isolated issues.

VI. Conclusion

Monetized cheering combines payments, social signaling, and real-time distributed systems, creating a distinctive abuse surface. Across identity, payments, wallets and ledgers, event streaming and aggregation, anomaly detection, and trust and safety operations, the analysis identifies five broad threat families: scalable identity manipulation, payment and ledger abuse (including chargeback and laundering-like patterns), replay or duplication of canonical events, denial-of-service pressure on latency-sensitive components, and misuse of privileged access in detection and enforcement tools. These threats interact: identity-driven spoofing can enable payment fraud, while compromised or misconfigured operational tooling can allow that fraud to go unnoticed. The practical implication is a layered protection strategy, with server-side event generation, replay resistance and idempotency, adaptive throttling and graceful degradation, tamper-evident audit and change control, privacy-aware measurement, and least-privilege access deployed at the trust boundaries where they are most effective.

The paper makes three contributions. First, it offers a reusable reference model for monetized cheering that makes trust boundaries and handoff risks explicit across identity, payments, event processing, and real-time social signaling. Second, it provides a STRIDE per-element threat register that connects scenarios to affected assets, trust-boundary crossings, and feasible controls, giving practitioners a repeatable starting point for platform-specific threat modelling. Third, it links that register to an ordinal DREAD prioritization, a worked multi-hypothesis anomaly example, and a privacy-security-fairness trade-off analysis. Together, these outputs show how anomaly signals can inform measured operational decisions and how established threat-modelling techniques can be adapted to real-time monetization systems.

The study has limitations. The reference architecture abstracts from platform-specific details such as settlement timing, overlay delivery, and operational workflows, so the threat register should be applied to a deployment-specific DFD rather than treated as a final specification. DREAD prioritization depends on attacker assumptions and organizational risk appetite, and the scores should remain ordinal and be revisited as controls and attacker capabilities change. The social and technical effects discussed here, including false positives, creator equity, and privacy implications, still need empirical measurement by channel size and community context. Future work should validate anomaly-to-threat mappings with labelled cases, quantify mitigation effectiveness under realistic traffic and payment conditions, extend the privacy analysis using LINDDUN, and develop attack-tree representations for multi-step laundering-like campaigns involving engagement markets and underground payment pathways.

Conflict of Interest

The authors declare no conflicts of interest.

Author Contributions

Conceptualization, methodology, formal analysis, and original draft preparation: E.E. Supervision, conceptualization, validation, review, editing, and manuscript refinement: S.F. All authors had approved the final version.

References

[1] W. A. Hamilton, O. Garretson, and A. Kerne, "Streaming on Twitch: Fostering participatory communities of play within live mixed media," in *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI'14)*, New York, 2014, pp. 1315–1324. doi: 10.1145/2556288.2557048

[2] Z. Hilvert-Bruce, J. T. Neill, M. Sjöblom, and J. Hamari, "Social motivations of live-streaming viewer engagement on Twitch," *Computers in Human Behavior*, vol. 84, pp. 58–67, 2018. doi: 10.1016/j.chb.2018.02.013

[3] M. R. Johnson and J. Woodcock, "'And today's top donator is': How live streamers on Twitch.tv monetize and gamify their broadcasts," *Social Media + Society*, vol. 5, no. 4, 2019. doi: 10.1177/2056305119881694

[4] E. Eminov, S. Flowerday, and A. Morin, "Suspicious cheering with bits," *Issues in Information Systems*, vol. 26, no. 3, pp. 457–471, 2025. doi: 10.48009/3_iis_2025_137

[5] J. Cai, S. Chowdhury, H. Zhou, and D. Y. Wohn, "Hate raids on Twitch: Understanding real-time human-bot coordinated attacks in live streaming communities," in *Proc. ACM Hum.-Comput. Interact.,* 2023, vol. 7, no. CSCW2, 342. doi: 10.1145/3610191

[6] E. Ferrara, O. Varol, C. A. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016. doi: 10.1145/2818717

[7] D. Nevado-Catalán, S. Pastrana, N. Vallina-Rodríguez, and J. E. Tapiador, "An analysis of fake social media engagement services," *Computers & Security*, vol. 124, 103013, 2023. doi: 10.1016/j.cose.2022.103013

[8] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002. doi: 10.1214/ss/1042727940

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, 15, pp. 1–58, 2009. doi: 10.1145/1541880.1541882

[10] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011. doi: 10.1016/j.dss.2010.08.008

[11] X. Ding, N. Seleznev, S. Kumar, C. B. Bruss, and L. Akoglu, "From detection to action: A human-in-the-loop toolkit for anomaly reasoning and management," in *Proc. 4th ACM Int. Conf. AI in Finance (ICAIF '23)*, New York, 2023, pp. 279–287. doi: 10.1145/3604237.3626872

[12] A. Shostack. (2008). Experiences threat modeling at Microsoft. *MODSEC@MoDELS 2008*. [Online]. Availeble: https://shostack.org/files/papers/modsec08/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf

[13] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011. doi: 10.1007/s00766-010-0115-7

[14] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004. doi: 10.1145/997150.997156

[15] M. Levi and P. Reuter, "Money laundering," *Crime and Justice*, vol. 34, no. 1, pp. 289–375, 2006. doi: 10.1086/501508

[16] Financial Action Task Force. (Jun 2014). Virtual currencies: Key definitions and potential AML/CFT risks. *FATF*. [Online]. Available: https://www.fatf-gafi.org/content/dam/fatf/documents/

reports/Virtual-currency-key-definitions-and-potential-aml-cft-risks.pdf
[17] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996. doi: 10.1109/2.485845
[18] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, RFC 3161, RFC Editor, Aug. 2001. doi: 10.17487/RFC3161
[19] OWASP Foundation. (2023). OWASP Top 10 API Security Risks–2023. [Online]. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/
[20] OWASP Foundation. (2025). OWASP Application Security Verification Standard (ASVS) 5.0.0. [Online]. Available: https://owasp.org/www-project-application-security-verification-standard/